WHAT IS CLAIMED IS:

1. A system for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said computer system further comprising:

a first page table representative of the logical allocation of physical memory associated with said foreign binary code;

a second page table representative of the logical allocation of physical memory associated with binary translated code; and

means for maintaining said first and second page tables in response to changes generated by executing said binary translated code.

2. The system of claim 1 wherein said maintaining means further comprises a register dedicated to track code modification to maintain correspondence between foreign and binary translated host code flag, associated with each page of memory, to indicate modification of said page of memory.

3. The system of claim 1 further comprising a flag, associated with each page of memory, to indicate a change in a page in foreign virtual space.

4. The system of claim 1 further comprising means for restricting access to a page of memory in foreign virtual space.

5. The system of claim 4 wherein said restricting means comprises means for generating a trap.

6. The system of claim 1 wherein said first page table maps foreign code to a portion of physical memory.

7. The system of claim 6 wherein said first page table includes a first and second bit associated with each page of memory in foreign virtual space, said first bit indicating whether a page of memory in foreign virtual space is write protected and said second bit indicating a locked page of memory

8. The system of claim 7 further comprising means for determining the state of said first bit and invoking a page fault exception upon a change in page contents.

9. The system of claim 1 wherein said first page table includes a first and second bit associated with each page of host memory, said first bit indicating whether access to said page is restricted; said second bit indicated whether access to said page in said foreign virtual space is accessible only in a supervisor mode of operation.

10. A method for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said method comprising the steps of:

configuring a first and second virtual space;

storing foreign code in said first virtual space;

storing a plurality of translation processes in said second virtual space;

storing binary translated code in said second virtual space, said binary translated code comprising at least a portion of said foreign code in said first virtual space;

executing said binary translated code;

detecting a memory access that attempts to modify a memory location;

determining the status of the accessed memory location;

if the status permits access, modifying the memory location;

if the status does not permit access, generating a page fault exception.

11. The method of claim 10 further comprising the step of:

if the status indicates access is limited to code executing in an emulated target supervisor mode, determining if the computer system is executing in an emulated target supervisor mode; and

if the status indicates that the code is not executing in said emulated target supervisor mode, generating an exception.

12. The method of claim 11 further comprising the steps

if access is allowed, checking the address against a range of addresses; and

determining if the memory location to be accessed is currently in memory.

13. A method for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said method comprising the steps of:

configuring a first and second virtual space;

storing foreign code and data in said first virtual space;

storing binary translated code in said second virtual space, said binary translated code comprising at least a portion of said foreign code in said first virtual space;

defining a first page table to map said foreign code from said first virtual space to physical memory;

defining a second page table to map said binary translated code to said second virtual space;

associating information relating to said foreign code with said page table, said information determining if a portion of said foreign code and data in said virtual space may be accessed.

14. The method of claim 13 further comprising the step of using said information to determine if said foreign code comprises operating foreign system code.

15. The method of claim 14 further comprising the step of using said information to detect modification of said code and data in said foreign virtual space.

16. The method of claim 15 further comprising the step of invoking a binary translation process upon detection of modification of said code and data in said foreign virtual space to generate binary translated code corresponding to said code and data in said foreign virtual space.

17. The method of claim 13 further comprising the step of using said first page table to map a logical address of said foreign code to a compatibility region of physical memory in said host computer system.

18. A method for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said method comprising the steps of:

configuring a first and second virtual space;

storing foreign code and data in said first virtual space;

storing binary translated code in said second virtual space, said binary translated code comprising at least a portion of said foreign code in said first virtual space;

defining a first page table to map said foreign code from said first virtual space to physical memory;

defining a second page table to map said binary translated code to said second virtual space;

providing a support software layer, said software layer functioning as an operating system for controlling the binary translation of said foreign code and data and for detecting a write access to physical memory of said host computer system, said physical memory corresponding to said foreign virtual space by said host code.

19. The method of claim 18 further comprising the step of executing said support software layer in said second virtual memory, said support software layer adapted to managing I/O operations.

20. The method of claim 19, further comprising the step of virtualizing a restricted set of functions.

21. The system of claim 20, further comprising the step of virtualizing peripheral components expected in a target operating system.

22. A system for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said computer system further comprising:

a first page table representative of the logical allocation of physical memory associated with said foreign binary code;

a second page table representative of the logical allocation of physical memory associated with binary translated code; and

a support software layer adapted to maintain correspondence between foreign code and data in said foreign virtual space and a set of host instructions in said host virtual space.

23. The system of claim 22 wherein said support software layer comprises means for performing binary translation of said foreign code and said host code.

24. The system of claim 23 further comprising an access watch engine adapted to detect a write access to physical memory in said host computer system.

25. The system of claim 22 further comprising means for detecting write accesses to physical memory corresponding to said foreign virtual space.

26. The system of claim 25 further comprising means for detecting write accesses to physical memory corresponding to memory mapped peripherals.

27. The system of claim 25 further comprising means for virtualizing a restricted set of available hardware to produce enough resources to run resource-consuming firmware.

28. A system for translating foreign binary code for execution on a host computer system where the host computer system is architecturally distinct from the foreign architecture, said host computer system providing a foreign and a host virtual space, said computer system further comprising:

a physical memory;

a first page table representative of the logical allocation of physical memory associated with said foreign binary code;

a second page table representative of the logical allocation of physical memory associated with binary translated code;

a support software layer adapted to maintain correspondence between foreign code and data in said foreign virtual space and a set of host instructions in said host virtual space; and

an emulated supervisor flag associated with portions of said host binary code where said support software layer accesses said supervisor flag prior to executing a write access into said foreign and host virtual spaces.

29. The system of claim 28 further comprising means for detecting said supervisor flag and marking selected portions of said foreign and host virtual spaces as accessible only in an emulated supervisor mode, said detecting means coupled to said support software layer.

30. The system of claim 28 further comprising a page lock flag, said page lock flag indicating restricted access to selected portions of said foreign virtual space.

31. The system of claim 30 further comprising means for detecting the state of said page lock flag.

32. The system of claim 28 further comprising a common structure supported in hardware for maintaining said the foreign and host virtual spaces.

33. The system of claim 28 further comprising an access watch engine for detecting write accesses to said physical memory.

34. The system of claim 28 further comprising an emulated target supervisor flag for initiating binary translation from said foreign binary code to said host binary code.

09238550.041801